VI International Conference on Computational Methods in Marine Engineering
MARINE 2015
F. Salvatore, R. Broglia and R. Muscari (Eds)

ADAPTIVE SIMULATION OF UNSTEADY FLOW PAST THE SUBMERGED PART OF A FLOATING WIND TURBINE PLATFORM

Johan Jansson*[12], Vincenzo Nava[3], Miren Sanchez[3], Goren Aguirre[3], Rodrigo Vilela de Abreu[2], Johan Hoffman[21] and Jose Luis Villate[3]

* Corresponding author: jjan@kth.se
[1] Basque Center for Applied Mathematics, Mazarredo 14., E-48009 Bilbao, Spain, bcamath.org
[2] Department of High Performance Computing and Visualization (HPCViz), KTH Royal Institute of Technology, SE-10044 Stockholm, Sweden ctl.csc.kth.se
[3] Energy and environment division, Tecnalia, c/ Geldo, edificio 700, E-48160, Derio-Bizkaia, Spain tecnalia.com

Key words: Adaptive, FEM, turbulence, Marine Engineering

Abstract.

Offshore floating platforms for wind turbines represent challenging concepts for designers trying to combine an optimal compromise between cost effectiveness and performance. Modelling of the hydrodynamic behaviour of the structure is still the subject of wide debate in the technical communities.

The assessment of the hydrodynamics of the support structure is not an easy task as the floaters consist of an assembly of columns, braces and pontoons, commonly also with heave plates: each of these components corresponds to a different hydrodynamic model and it further interacts with the other elements. This results in very complex non-linear modeling, which makes it necessary to resort to computational fluid dynamics (CFD) methods for the evaluation of the combined hydrodynamics.

In the framework of the collaboration between the Basque Centre for Applied Mathematics (BCAM) and Tecnalia R&I, the interaction of the sea flow with a semisubmersible floating offshore wind platform have been calculated by using the open source solver Unicorn in the FEniCS-HPC framework when subject to a steady inflow. The prototype of the platform consists in a semi-submersible 4-columns column stabilized platform -NAUTILUS Floating Solutions concept-; columns are connected by a rigid ring pontoon provided with heave damping plates at the bottom. The novelty of the approach in FEniCS-HPC hinges upon an implicit formulation for the turbulence, a cheap free slip model of the boundary layer and goal-oriented mesh adaptivity [8, 6, 9, 20, 1]. We find that the results are consistent with experimental results for cylinders at high Reynolds number.

1 Introduction

In this paper we present a time-resolved, adaptive finite element method without turbulence modeling parameters for aero- and hydrodynamics, together with simulation results for the submerged part of a floating wind turbine platform, illustrated in Figure 1. This work is a collaboration between the Basque Centre for Applied Mathematics (BCAM) and Tecnalia R&I to computationally model the interaction of sea flow with a semisubmersible floating offshore wind platform.

Due to the computational cost it is still today a computational challenge to accurately and efficiently compute aero- and hydrodynamic forces (e.g. drag) for high Reynolds numbers [22]. The ability to perform such simulations at an affordable cost could greatly enhance the virtual design process for marine structures, wind turbines, propellers, airplanes, etc.

Direct numerical simulation (DNS) of turbulent flow is limited to moderate Reynolds numbers and simple geometry, due to the high computational cost of resolving all turbulent scales in the flow. In the case of airfoils DNS is limited to laminar boundary layers since computational resolution of turbulent boundary layers is too expensive, see e.g. [27].

The standard e.g. in the automotive and aircraft industries has been simulation based on Reynolds averaged Navier-Stokes equations (RANS), where time averages (or statistical averages) are computed to an affordable cost, with the drawback of introducing turbulence models based on parameters that have to be tuned for particular applications.

An alternative to DNS and RANS is Large eddy simulation (LES) [26], where only the largest scales of the flow are resolved, combined with subgrid models to take into consideration the effect of the smallest unresolved turbulent scales. Subgrid models are either explicit, based on physics theory or experiments, or implicit through the numerical discretization of the equations.

A key challenge for LES methods is the modeling of turbulent boundary layers at different angles of attack. Full resolution of turbulent boundary layers is not feasible, due to the high cost associated with computational representation of all the physical scales. Instead cheaper models are used, including resolution of the boundary layer only in the wall-normal direction, wall shear stress models, and hybrid LES-RANS models such as DES [23, 17, 29, 24, 21].

The basis for our new methodology is an adaptive finite element method without boundary layer resolution. The mesh is automatically constructed by the method as part of the computation through duality-based a posterori error control, and no explicit turbulence model is needed. Dissipation of turbulent kinetic energy in under-resolved parts of the flow is provided by the numerical stabilization in the form of a weighted least squares method based on the residual of the Navier-Stokes Equations (NSE). Thus, the method is purely based on the NSE mathematical model, and no other modeling assumptions are made.

The effect of unresolved turbulent boundary layers is modeled by a simple parametriza-

tion of the wall shear stress in terms of the skin friction. In the case of very high Reynolds numbers (Re) we approximate the small skin friction by zero skin friction, corresponding to a free slip boundary condition, which results in a computational model without any model parameters that need tuning. Thus, the simulation methodology bypasses the main challenges posed by high Re CFD: the design of an optimal computational mesh, turbulence (or subgrid) modeling, and the cost of boundary layer resolution.

In this paper we present the main components of the simulation methodology and our drag results for the floating platform. We highlight the non-standard aspects of the methodology and discuss the results in relation to experiments for cylinders at high Reynolds numbers, which we argue should give comparable results since the platform is composed of four cylinders in each corner of a plate, where the cylinders comprise the majority of the projected area.

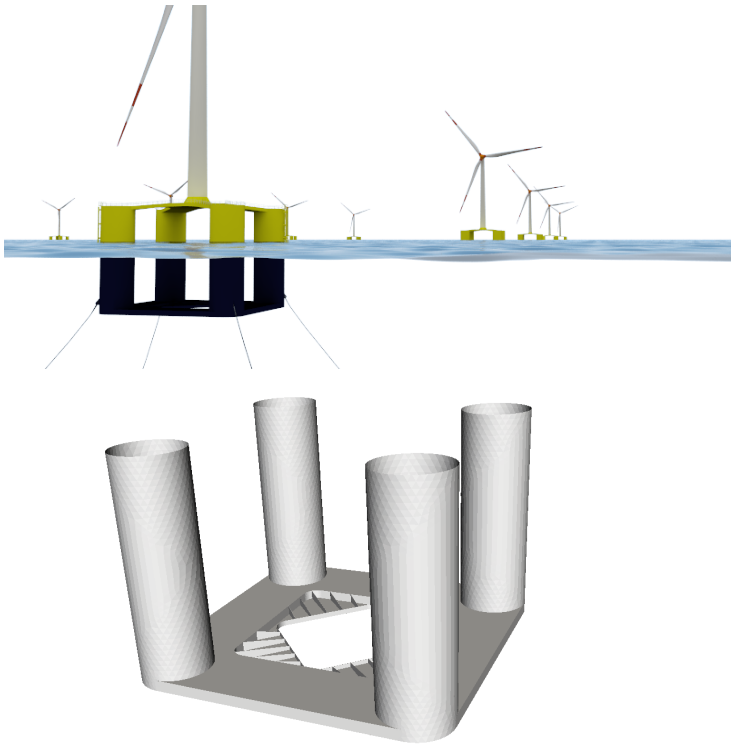

Figure 1: Illustration of the full NAUTILUS platform (top) and geometry of the submerged part (bottom).

2 Simulation Methodology

The mathematical framework for the simulation method is functional analysis and the concept of weak solutions to the Navier-Stokes equations (NSE), introduced by the

mathematician Jean Leray in 1934. Leray proved that there exist weak solutions (or turbulent solutions in the terminology of Leray) that satisfy NSE in variational form, that is NSE integrated against a family of test functions.

A finite element method (FEM) is based on the variational form of NSE, and one can show that, if the formulation of the method satisfies certain conditions on stability and consistency, the approximate FEM solutions converge towards a weak solution of the NSE as the finite element mesh is refined [13]. We refer to such FEM as General Galerkin (G2) methods.

The test functions in G2 are defined over the mesh, and thus the finest scales of a G2 approximation are set by the mesh size. In contrast to RANS or LES (Large eddy simulation), no averaging operator or filter is applied to NSE, and thus no Reynolds or subgrid stresses that need modeling are introduced. Dissipation of turbulent kinetic energy in under-resolved parts of the flow is provided by the numerical stabilization of G2 in the form of a weighted least squares method based on the residual of NSE. Thus, the method is purely based on the NSE mathematical model, and no other modeling assumptions are made.

In G2, the mesh is adaptively constructed based on *a posteriori* estimation of the error in chosen goal or target functionals, such as drag and lift forces for example. Using duality in a variational framework, *a posteriori* error estimates can be derived in terms of the residual, the mesh size, and the solution of a "dual" (or "adjoint") problem [15]. We initiate the adaptive mesh refinement algorithm from a coarse mesh, fine enough to capture the geometry, but without any further assumptions on the solution (i.e., no boundary layer meshes or *ad hoc* mesh design based on expected separation and wake structures are needed).

To model the effect of unresolved turbulent boundary layers we use a simple parametrization of the wall shear stress in terms of the skin friction. In particular, for the very high Re for this problem we approximate the small skin friction by zero skin friction, which corresponds to a free slip boundary condition without boundary layer resolution.

This methodology is validated for a number of standard benchmark problems in the literature [3, 4, 14, 5], and in the following sections we describe the basic elements of the G2 method, also referred to as Adaptive DNS/LES, or simply Direct Finite element Simulation (DFS).

For this particular problem of the floating platform, we have used a low order finite element discretization on unstructured tetrahedral meshes, which we refer to as cG(1)cG(1), i.e., continuous piecewise linear approximations in space and time.

2.1 The cG(1)cG(1) method

In a cG(1)cG(1) method [13] we seek an approximate space-time solution $\hat{U} = (U, P)$ which is continuous piecewise linear in space and time (equivalent to the implicit Crank-Nicolson method). With I a time interval with subintervals $I_n = (t_{n-1}, t_n)$, W^n a standard spatial finite element space of continuous piecewise linear functions, and W_0^n the functions

in W^n which are zero on the boundary Γ, the cG(1)cG(1) method for constant density incompressible flow with homogeneous Dirichlet boundary conditions for the velocity takes the form: for $n = 1, ..., N$, find $(U^n, P^n) \equiv (U(t_n), P(t_n))$ with $U^n \in V_0^n \equiv [W_0^n]^3$ and $P^n \in W^n$, such that

$$\begin{aligned} r(\hat{U}, \hat{v}) &= ((U^n - U^{n-1})k_n^{-1} + (\bar{U}^n \cdot \nabla)\bar{U}^n, v) + (2\nu\epsilon(\bar{U}^n), \epsilon(v)) \\ &- (P, \nabla \cdot v) + (\nabla \cdot \bar{U}^n, q) + LS(U, P) = 0, \ \forall \hat{v} = (v, q) \in V_0^n \times W^n \end{aligned} \tag{1}$$

where $\bar{U}^n = 1/2(U^n + U^{n-1})$ is piecewise constant in time over I_n and LS a least-squares stabilizing term described in [13].

2.2 A posteriori error estimate for cG(1)cG(1)

We formulate an adjoint-based method for adaptive error control based on the following error representation and adjoint weak bilinear and linear forms with the error $\hat{e} = \hat{u} - \hat{U}$, adjoint solution $\hat{\phi}$ and the hat signifying the full velocity-pressure vector $\hat{U} = (U, P)$:

$$(\hat{e}, \psi) = \overline{r'}(\hat{e}, \hat{\phi}) = r(\hat{U}; \hat{\phi}) \qquad a_{adjoint}(v, \hat{\phi}) = \overline{r'}(v, \hat{\phi}) \qquad L_{adjoint}(v) = (v, \psi) \tag{2}$$

$$a_{adjoint}(v, \hat{\phi}) = L_{adjoint}(v) = (v, \psi), \quad \forall \hat{v} \in V_0^n \tag{3}$$

The *a posteriori* error estimate is based on the following theorem (for a detailed proof, see chapter 30 in [15]):

Theorem 1 *If $\hat{U} = (U, P)$ solves (1), $\hat{u} = (u, p)$ is a weak NSE solution, and $\hat{\varphi} = (\varphi, \theta)$ solves an associated dual problem with data $M(\cdot)$, then we have the following a posteriori error estimate for the target functional $M(\hat{U})$ with respect to the reference functional $M(\hat{u})$:*

$$\begin{aligned} |M(\hat{u}) - M(\hat{U})| \leq \sum_{n=1}^{N} [\ & \int_{I_n} \sum_{K \in \mathcal{T}_n} |R_1(\hat{U})|_K \cdot \omega_1 \ dt \\ + \int_{I_n} \sum_{K \in \mathcal{T}_n} |R_2(\hat{U})|_K \ \omega_2 \ dt &+ \int_{I_n} \sum_{K \in \mathcal{T}_n} |LS_\delta^n(\hat{U}; \hat{\varphi})_K| \ dt \ ] =: \sum_{K \in \mathcal{T}_n} \mathcal{E}_K \end{aligned}$$

with

$$\begin{aligned} R_1(\hat{U}, P) &= \dot{U} + (U \cdot \nabla)U + \nabla P - 2\nu\nabla \cdot \epsilon(u) - f, \\ R_2(U) &= \nabla \cdot U, \end{aligned} \tag{4}$$

where $LS_\delta^n(\cdot;\cdot)_K$ is a local version of the stabilization, and the stability weights are given by

$$\begin{aligned} \omega_1 &= C_1 h_K |\nabla\varphi|_K, \\ \omega_2 &= C_2 h_K |\nabla\theta|_K, \end{aligned}$$

where h_K is the diameter of element K in the mesh $\mathcal{T}_k$, and $C_{1,2}$ represent interpolation constants. Moreover, $|w|_K \equiv (\|w_1\|_K, \|w_2\|_K, \|w_3\|_K)$, with $\|w\|_K = (w, w)_K^{1/2}$, and the dot denotes the scalar product in $\mathbb{R}^3$.

2.3 The Adaptive Algorithm

A simple description of the adaptive algorithm, starting from $k = 0$, reads:

1. For the mesh $\mathcal{T}_k$: compute primal and (linearized) dual problems for the primal solution (U, P) and dual solution (Φ, Θ).
2. If $\sum_{K\in\mathcal{T}_k} \mathcal{E}_K < TOL$ then stop, else:
3. Mark 10% of the elements with highest $\mathcal{E}_K$ for refinement.
4. Generate the refined mesh $\mathcal{T}_{k+1}$, and goto 1.

The formulation of the dual problem includes the definition of a *target functional* for the refinement, which usually enters the dual equations as a boundary condition or as a volume source term. This functional should be chosen according to the problem we are solving. In other words, one needs to ask the right question in order to obtain the correct answer from the algorithm. In this paper our target functional is chosen to be the mean value in time of the drag force.

The only other input required from the user is an initial discretization of the geometry, $\mathcal{T}_0$. Since our method is designed for tetrahedral meshes that do not require any special treatment of the near wall region (no need for a boundary-layer mesh), the initial mesh can be easily created with any standard mesh generation tool.

As mentioned above, the error indicator, $\mathcal{E}_K$, is a function of the residual of the NSE and the solution of a linearized dual problem. Thus, on a given mesh, we must first solve the NSE to compute the residual, and then a linearized dual problem to compute the weights multiplying the residuals. With that information, we are able to compute $\sum_{K\in\mathcal{T}_k} \mathcal{E}_K$ and check it against the given stop criterion. This procedure of solving the forward and backward problems for the NSE is closely related to an optimization loop and can be understood as the problem of finding the "optimal mesh" for a given geometry and boundary conditions, *i.e.*, the mesh with the least possible number of degrees of freedom for computing $M(\hat{u})$ within a given degree of accuracy.

2.4 Turbulent boundary layers

In our work on high Reynolds number turbulent flow [12, 16, 30], we have chosen to apply a skin friction stress as wall layer model. That is, we append the NSE with the following boundary conditions:

$$u \cdot n = 0, \tag{5}$$

$$\beta u \cdot \tau_k + n^T \sigma \tau_k = 0, \quad k = 1, 2, \tag{6}$$

for $(x,t) \in \Gamma_{solid} \times I$, with $n = n(x)$ an outward unit normal vector, and $\tau_k = \tau_k(x)$ orthogonal unit tangent vectors of the solid boundary Γ_{solid}. We use matrix notation with all vectors v being column vectors and the corresponding row vector is denoted v^T.

With skin friction boundary conditions, the rate of kinetic energy dissipation in cG(1)cG(1) has a contribution of the form:

$$\sum_{k=1}^{2} \int_0^T \int_{\Gamma_{solid}} |\beta^{1/2} \bar{U} \cdot \tau_k|^2 \, ds \, dt, \tag{7}$$

from the kinetic energy which is dissipated as friction in the boundary layer. For high Re, we model $Re \to \infty$ by $\beta \to 0$, so that the dissipative effect of the boundary layer vanishes with large Re. In particular, we have found that a small β does not influence the solution [12]. For the present simulations we used the approximation $\beta = 0$, which can be expected to be a good approximation for high Reynolds numbers.

2.5 The FEniCS-HPC finite element computational framework

The simulations in this article have been computed using the Unicorn solver in the FEniCS-HPC automated FEM software framework.

FEniCS-HPC is an open source framework for automated solution of PDE on massively parallel architectures, providing automated evaluation of variational forms given a high-level description in mathematical notation, duality-based adaptive error control, implicit parameter-free turbulence modeling by use of stabilized FEM and strong linear scaling up to thousands of cores [6, 7, 18, 19, 11, 10]. FEniCS-HPC is a branch of the FEniCS [20, 1] framework focusing on high performance on massively parallel architectures.

Unicorn is solver technology (models, methods, algorithms and software) with the goal of automated high performance simulation of realistic continuum mechanics applications, such as drag or lift computation for fixed or flexible objects (FSI) in turbulent incompressible or compressible flow. The basis for Unicorn is Unified Continuum (UC) modeling [2] formulated in Euler (laboratory) coordinates, together with the General Galerkin (G2) adaptive stabilized finite element discretization described above.

The simulations in this paper were run on supercomputer resources described in the Acknowledgments section, and took ca. 48h on the finest mesh for the whole time interval using ca. 1000 cores.

3 Test problem description

The geometry analyzed in the present work consists of a four column-stabilized platform for offshore wind purposes (NAUTILUS Floating Solutions concept). It is a four column ring pontoon semi-submersible unit with heave plates and a catenary mooring system, see Figure 1 for an illustration. The wind turbine is located centered relative to the columns. In this work only the submerged part below the mean water level of the geometry is considered, see Figure 1.

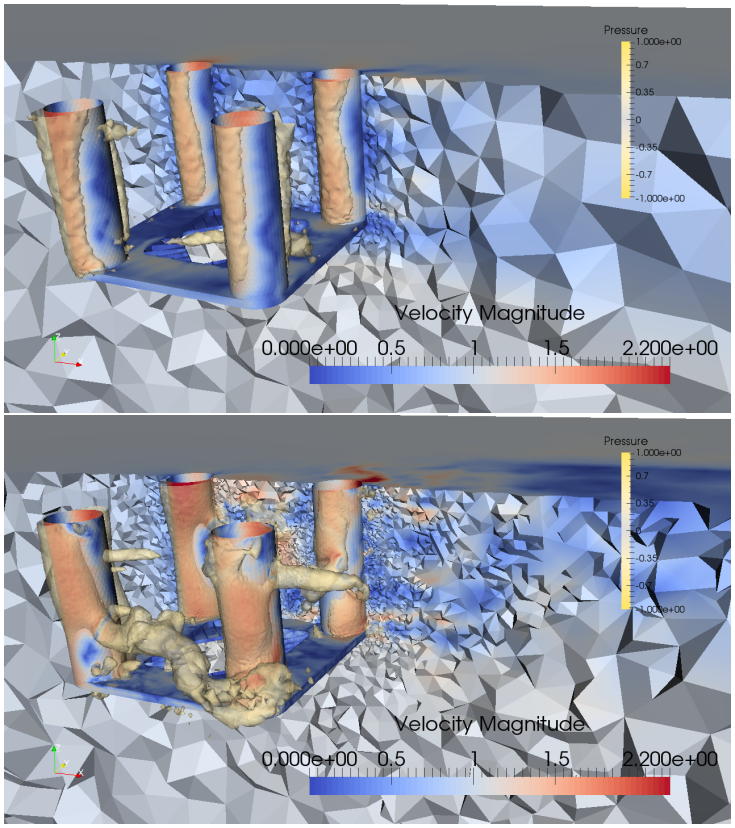

Figure 2: Solution on coarse initial mesh (top) and finest adaptive mesh (bottom).

The four columns, 9.5 m diameter each and placed in a square pattern at a distance of 33 m one another, provide buoyancy to support the turbine and enough water plane inertia for the stability. The columns are connected by a rigid ring pontoon provided with heave damping plates at the bottom. The horizontal plates at the bottom and in between the columns increase the added mass; hence, shift the natural period away from the wave energy, and increase the viscous damping in roll, pitch, and heave. The operational draft is around 20 m.

4 Results

We compute adaptive time-dependent solutions to the incompressible Euler equations in non-dimensional form with a unit inflow and take the time-average of the drag force. Each adaptive iteration gives a velocity and pressure, and we plot the solution on the initial coarse mesh and the finest adaptive mesh in Figure 2. From each adaptive iteration we also generate a dual velocity and pressure giving sensititvity information about the computational error, which are used to refine the mesh based on an error estimate. We plot the dual velocity on the finest mesh in Figure 3).

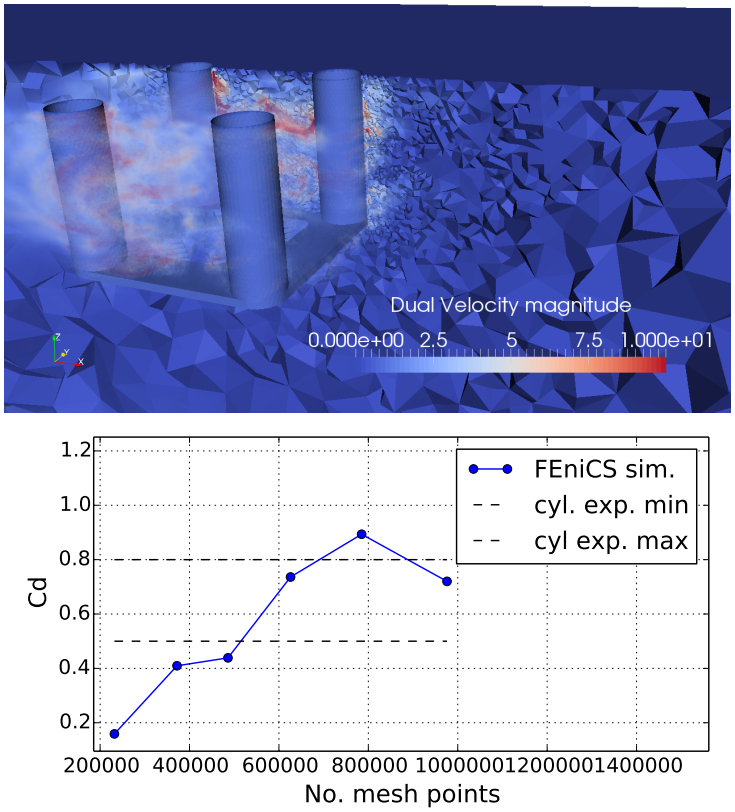

Figure 3: Dual solution (sensititvity weight in error estimate) (top) and mesh convergence plot comparing against experimental values for a cylinder at high Re (bottom).

The predicted drag coefficient Cd on the finest mesh is 0.72, with the adaptivity indicating an error interval of ca. 25%. We plot the evolution of Cd on the adaptive meshes in Figure 3. We compare against experiments for cylinders at comparable Reynolds numbers [28, 25]. We see that the computed drag is slightly higher the experimental values for a cylinder, which is what we expect given the additional contribution from the plate.

5 Conclusions

We find that the early simulation results are consistent with the experimental results for cylinders, with a slightly higher drag explained by the presence of the plate. The mesh convergence study that is an integrated part of the iterative adaptive mesh refinement algorithm indicates that the drag has stabilized at a high value, but that the computational error is still relatively large: within an interval of ca. 25%. A clear direction for future work is thus a more detailed computational study with more adaptive refinement iterations.

Acknowledgments

This research has been supported by EU-FET grant EUNISON 308874, the European Research Council, the Swedish Foundation for Strategic Research, the Swedish Research Council, the Basque Excellence Research Center (BERC 2014-2017) program by the Basque Government, the Spanish Ministry of Economy and Competitiveness MINECO: BCAM Severo Ochoa accreditation SEV-2013-0323 and the Project of the Spanish Ministry of Economy and Competitiveness with reference MTM2013-40824.

We acknowledge PRACE for awarding us access to the supercomputer resources Hermit and SuperMUC based in Germany at The High Performance Computing Center Stuttgart (HLRS) and Leibniz Supercomputing Center (LRZ), from the Swedish National Infrastructure for Computing (SNIC) at PDC – Center for High-Performance Computing and on resources provided by the "Red Española de Supercomputación" and the "Barcelona Supercomputing Center - Centro Nacional de Supercomputación".

REFERENCES

[1] FEniCS. Fenics project. *http://www.fenicsproject.org*, 2003.

[2] J. Hoffman, J. Jansson, and M. Stöckli. Unified continuum modeling of fluid-structure interaction. *Mathematical Models and Methods in Applied Sciences*, 2011.

[3] Johan Hoffman. Computation of mean drag for bluff body problems using adaptive dns/les. *SIAM J. Sci. Comput.*, 27(1):184–207, 2005.

[4] Johan Hoffman. Adaptive simulation of the turbulent flow past a sphere. *J. Fluid Mech.*, 568:77–88, 2006.

[5] Johan Hoffman. Efficient computation of mean drag for the subcritical flow past a circular cylinder using general galerkin g2. *Int. J. Numer. Meth. Fluids*, 59(11):1241–1258, 2009.

[6] Johan Hoffman, Johan Jansson, Rodrigo Vilela de Abreu, Niyazi Cem Degirmenci, Niclas Jansson, Kaspar Müller, Murtazo Nazarov, and Jeannette Hiromi Spühler. Unicorn: Parallel adaptive finite element simulation of turbulent flow and fluid-structure interaction for deforming domains and complex geometry. *Computers and Fluids*, 2012.

[7] Johan Hoffman, Johan Jansson, Cem Degirmenci, Niclas Jansson, and Murtazo Nazarov. *Unicorn: a Unified Continuum Mechanics Solver*, chapter 18. Springer, 2012.

[8] Johan Hoffman, Johan Jansson, Niclas Jansson, and Rodrigo Vilela De Abreu. Towards a parameter-free method for high reynolds number turbulent flow simulation

based on adaptive finite element approximation. *Computer Methods in Applied Mechanics and Engineering*, 288(0):60 – 74, 2015. Error Estimation and Adaptivity for Nonlinear and Time-Dependent Problems.

[9] Johan Hoffman, Johan Jansson, Niclas Jansson, and Rodrigo Vilela de Abreu. Time-resolved adaptive fem simulation of the dlr-f11 aircraft model at high reynolds number. *AIAA SciTech 2014 Conference Proceedings*, 2014.

[10] Johan Hoffman, Johan Jansson, Niclas Jansson, C. Johnson, and Rodrigo V. de Abreu. Turbulent flow and fluid-structure interaction. In *Automated Solutions of Differential Equations by the Finite Element Method*. Springer, 2011.

[11] Johan Hoffman, Johan Jansson, Niclas Jansson, and Murtazo Nazarov. Unicorn: A unified continuum mechanics solver. In *Automated Solutions of Differential Equations by the Finite Element Method*. Springer, 2011.

[12] Johan Hoffman and Niclas Jansson. *A computational study of turbulent flow separation for a circular cylinder using skin friction boundary conditions*. Ercoftac, series Vol.16, Springer, 2010.

[13] Johan Hoffman and Claes Johnson. *Computational Turbulent Incompressible Flow: Applied Mathematics Body and Soul Vol 4*. Springer-Verlag Publishing, 2006.

[14] Johan Hoffman and Claes Johnson. A new approach to computational turbulence modeling. *Comput. Methods Appl. Mech. Engrg.*, 195:2865–2880, 2006.

[15] Johan Hoffman and Claes Johnson. *Computational Turbulent Incompressible Flow*, volume 4 of *Applied Mathematics: Body and Soul*. Springer, 2007.

[16] Johan Hoffman and Claes Johnson. Resolution of d'alembert's paradox. *J. Math. Fluid Mech.*, Published Online First at www.springerlink.com: 10 December 2008.

[17] L. Huang, P. G. Huang, and R. P. LeBeau. Numerical study of blowing and suction control mechanism on naca 0012 airfoil. *AIAA Journal of aircraft*, 41(1), 2004.

[18] Niclas Jansson, Johan Hoffman, and Johan Jansson. Framework for Massively Parallel Adaptive Finite Element Computational Fluid Dynamics on Tetrahedral Meshes. *SIAM J. Sci. Comput.*, 34(1):C24–C41, 2012.

[19] Robert C. Kirby. *FIAT: Numerical Construction of Finite Element Basis Functions,*, chapter 13. Springer, 2012.

[20] Anders Logg, Kent-Andre Mardal, Garth N. Wells, et al. *Automated Solution of Differential Equations by the Finite Element Method*. Springer, 2012.

[21] C. P. Mellen, J. Frölich, and W. Rodi. Lessons from lesfoil project on large-eddy simulation of flow around an airfoil. *AIAA journal*, 41:573–581, 2003.

[22] P. Moin and J. Kim. Tackling turbulence with supercomputers. 276(1):62–68, 1997.

[23] P. Moin and D. You. Active control of flow separation over an airfoil using synthetic jets. *Journal of Fluids and Structures*, 24(8):1349–1357, 2008.

[24] U. Piomelli and E. Balaras. Wall-layer models for large-eddy simulation. *Annu. Rev. Fluid Mech.*, 34:349–374, 2002.

[25] Anatol Roshko. Experiments on the flow past a circular cylinder at very high reynolds number. *Journal of Fluid Mechanics*, 10(03):345–356, 1961.

[26] P. Sagaut. *Large Eddy Simulation for Incompressible Flows (3rd Ed.)*. Springer-Verlag, Berlin, Heidelberg, New York, 2005.

[27] Hua Shan, Li Jiang, and Chaoqun Liu. Direct numerical simulation of flow separation around a naca 0012 airfoil. *Computers and Fluids*, 34:1096–1114, 2005.

[28] WCL Shih, C Wang, D Coles, and A Roshko. Experiments on flow past rough circular cylinders at large reynolds numbers. *Journal of Wind Engineering and Industrial Aerodynamics*, 49(1):351–368, 1993.

[29] P. R. Spalart. Detached-eddy simulation. *Annu Rev. Fluid Mech.*, 41:181–202, 2009.

[30] Rodrigo Vilela de Abreu, Niclas Jansson, and Johan Hoffman. Adaptive computation of aeroacoustic sources for a rudimentary landing gear. *Int. J. Numer. Meth. Fluids*, 74(6):406–421, 2014.